

**What Is Claimed Is:**

1           1.       A method for protecting a server against denial-of-service attacks,  
2       comprising:  
3           receiving a request for service at the server, wherein the request is received  
4       from a client;  
5           in response to the request, sending a random number,  $y$ , and an identifier,  
6        $id_1$ , to the client;  
7           allowing the client to compute a preimage,  $x$ , such that  $y = h(x)$ ;  
8           receiving an answer from the client, including the preimage  $x$  and an  
9       identifier,  $id_2$ ;  
10          verifying that the identifier,  $id_1$ , sent to the client matches the identifier,  
11        $id_2$ , received from the client;  
12          if the identifiers match, computing  $h(x)$ ; and  
13          if  $h(x) = y$ , performing the requested service for the client;  
14          whereby the server avoids computing  $h(x)$  until the server receives the  
15       answer with a matching identifier.

1           2.       The method of claim 1, wherein the server sends a parameter,  $n$ ,  
2       along with the random number  $y$  to the client, wherein the parameter  $n$  varies the  
3       amount of computational work involved in computing the preimage  $x$ .

1           3.       The method of claim 2, wherein the parameter  $n$  specifies that a  
2       subset of  $n$  bits of  $h(x)$  has to match a corresponding subset of  $n$  bits of  $y$ .

1           4.       The method of claim 1, wherein computing the preimage,  $x$ , takes  
2       more computational effort than computing  $h(x)$ , whereby the client is forced to

3 perform more computational work than the server before the server performs the  
4 requested service.

1 5. The method of claim 1, wherein if  $y \neq h(x)$ , the server ignores  
2 subsequent communications from the client.

1 6. The method of claim 1, wherein if  $y \neq h(x)$ , the server becomes  
2 slower in responding to subsequent communications from the client, distinguished  
3 from other clients, as by its source IP address.

1 7. The method of claim 6, wherein each time the server determines  
2  $y \neq h(x)$ , the server doubles the service time for the client, distinguished from  
3 other clients, as by its source IP address, so that the server spends progressively  
4 less time servicing requests for the client.

1 8. The method of claim 1,  
2 wherein sending the random number,  $y$ , and the identifier,  $id_1$ , to the client  
3 involves first,  
4 generating the random number  $y$  and the identifier  $id_1$ ; and  
5 storing the random number  $y$  and the identifier  $id_1$  at the  
6 server; and  
7 wherein verifying that  $id_1$  matches  $id_2$  involves first looking up  $id_1$  and the  
8 random number  $y$  at the server.

1 9. The method of claim 1, wherein  $h(x)$  is a hash function.

1           10.     The method of claim 1, wherein the identifier,  $id_1$ , is inferred from  
2 data related to the communication.

1           11.     A computer-readable storage medium storing instructions that  
2 when executed by a computer cause the computer to perform a method for  
3 protecting a server against denial-of-service attacks, the method comprising:  
4           receiving a request for service at the server, wherein the request is received  
5 from a client;  
6           in response to the request, sending a random number,  $y$ , and an identifier,  
7  $id_1$ , to the client;  
8           allowing the client to compute a preimage,  $x$ , such that  $y = h(x)$ ;  
9           receiving an answer from the client, including the preimage  $x$  and an  
10 identifier,  $id_2$ ;  
11           verifying that the identifier,  $id_1$ , sent to the client matches the identifier,  
12  $id_2$ , received from the client;  
13           if the identifiers match, computing  $h(x)$ ; and  
14           if  $h(x) = y$ , performing the requested service for the client;  
15           whereby the server avoids computing  $h(x)$  until the server receives the  
16 answer with a matching identifier.

1           12.     The computer-readable storage medium of claim 11, wherein the  
2 server sends a parameter,  $n$ , along with the random number  $y$  to the client, wherein  
3 the parameter  $n$  varies the amount of computational work involved in computing  
4 the preimage  $x$ .

1           13.     The computer-readable storage medium of claim 11, wherein the  
2     parameter  $n$  specifies that a subset of  $n$  bits of  $h(x)$  has to match a corresponding  
3     subset of  $n$  bits of  $y$ .

1           14.     The computer-readable storage medium of claim 11, wherein  
2     computing the preimage,  $x$ , takes more computational effort than computing  $h(x)$ ,  
3     whereby the client is forced to perform more computational work than the server  
4     before the server performs the requested service.

1           15.     The computer-readable storage medium of claim 11, wherein if  
2      $y \neq h(x)$ , the server ignores subsequent communications from the client.

1           16.     The computer-readable storage medium of claim 11, wherein if  
2      $y \neq h(x)$ , the server becomes slower in responding to subsequent communications  
3     from the client, distinguished from other clients, as by its source IP address.

1           17.     The computer-readable storage medium of claim 16, wherein each  
2     time the server determines  $y \neq h(x)$ , the server doubles the service time for the  
3     client, distinguished from other clients, as by its source IP address, so that the  
4     server spends progressively less time servicing requests for the client.

1           18.     The computer-readable storage medium of claim 11,  
2     wherein sending the random number,  $y$ , and the identifier,  $id_i$ , to the client  
3     involves first,  
4                     generating the random number  $y$  and the identifier  $id_i$ ; and  
5                     storing the random number  $y$  and the identifier  $id_i$  at the  
6     server; and

7            wherein verifying that  $id_1$  matches  $id_2$  involves first looking up  $id_1$  and the  
8            random number  $y$  at the server.

1            19.    The computer-readable storage medium of claim 11, wherein  $h(x)$   
2            is a hash function.

1            20.    The computer-readable storage medium of claim 11, wherein the  
2            identifier,  $id_1$ , is inferred from data related to the communication.

1            21.    An apparatus that protects a server against denial-of-service  
2            attacks, comprising:  
3            the server;  
4            a receiving mechanism within the server that is configured to receive a  
5            request for service from a client;  
6            an access mechanism, wherein in response to the request, the access  
7            mechanism is configured to,  
8                      send a random number,  $y$ , and an identifier,  $id_1$ , to the  
9                      client,  
10                     allow the client to compute a preimage,  $x$ , such that  
11                      $y = h(x)$ ,  
12                     receive an answer from the client, including the preimage  $x$   
13                     and an identifier,  $id_2$ , and to  
14                     verify that the identifier,  $id_1$ , sent to the client matches the  
15                     identifier,  $id_2$ , received from the client,  
16            wherein if the identifiers match, the access mechanism is configured to  
17            compute  $h(x)$ ; and

18 wherein if  $h(x) = y$ , the server is configured to perform the requested  
19 service for the client;  
20 whereby the server avoids computing  $h(x)$  until the server receives the  
21 answer with a matching identifier.

1 22. The apparatus of claim 21, wherein the access mechanism is  
2 configured to send a parameter,  $n$ , along with the random number  $y$  to the client,  
3 wherein the parameter  $n$  varies the amount of computational work involved in  
4 computing the preimage  $x$ .

1 23. The apparatus of claim 22, wherein the parameter  $n$  specifies that a  
2 subset of  $n$  bits of  $h(x)$  has to match a corresponding subset of  $n$  bits of  $y$ .

1 24. The apparatus of claim 21, wherein computing the preimage,  $x$ ,  
2 takes more computational effort than computing  $h(x)$ , whereby the client is forced  
3 to perform more computational work than the server before the server performs  
4 the requested service.

1 25. The apparatus of claim 21, wherein if  $y \neq h(x)$ , the server is  
2 configured to ignore subsequent communications from the client.

1 26. The apparatus of claim 21, wherein if  $y \neq h(x)$ , the server is  
2 configured to become slower in responding to subsequent communications from  
3 the client, distinguished from other clients, as by its source IP address.

1 27. The apparatus of claim 26, wherein each time the server  
2 determines  $y \neq h(x)$ , the server is configured to double the service time for the

1 client, distinguished from other clients, as by its source IP address, so that the  
2 server spends progressively less time servicing requests for the client.

1 28. The apparatus of claim 21, wherein the access mechanism is  
2 additionally configured to:  
3 generate the random number  $y$  and the identifier  $id_I$ ;  
4 store the random number  $y$  and the identifier  $id_I$  at the server; and  
5 upon receiving the answer from the client, to look up  $id_I$  and the random  
6 number  $y$  at the server.

1 29. The apparatus of claim 21, wherein  $h(x)$  is a hash function.

1 30. The apparatus of claim 21, wherein the identifier,  $id_I$ , is inferred  
2 from data related to the communication.